Introdução

Hoje, os cursos da área de computação, Ciência da Computação, por exemplo, tem uma quantidade de meninos muito maior do que a quantidade de meninas. Motivada por essa estátistica, a Professora Maristela Terto de Holanda aplicou um formulário de 2011 a 2014 na Semana Nacional de Ciência e Tecnologia (SNCT) em Brasília buscando encontrar fatores que são impactantes quando um menina quer escolher um curso de computação. Os dados foram disponibilizados pelo professor Vinicius Ruela Pereira Borges.

Esse trabalho irá realizar as seguintes atividades:

1. **Limpeza dos Dados:** explicar como foi o processo de carregamento dos dados (o processo de extração foi previamente feito) e como os dados inconsistentes foram removidos.
2. **Classificação de Atributos:** explicar quais métodos foram usados para classificar a importância/seleção dos dados para a próxima etapa.
3. **Modelo de Treino**: usando o modelo *Support Vector Machine* [1], criar um modelo que mostre quais os atributos são mais importantes quando um menina vai decidir que quer ou não fazer o curso de Ciência da Computação.
4. **Análise exploratória:** buscando encontrar de forma manual pontos relevantes quando um menina vai decidir que quer ou não fazer o curso de Ciência da Computação.

Objetivo

Utilizar os dados disponíveis e a linguagem R para tentar encontrar fatores que tenham correlação e/ou influência quando meninas decidem qual curso superior querem fazer e se vão ou não fazer um curso na área de computação.

Pacotes Utilizados

```
library(dplyr)
library(caret)
library(corrplot)
library(knitr)
library(ggplot2)
library(Rmisc)
```

1. Limpeza dos Dados

Inicialmente, o *data frame* possui as seguintes informações armazenadas.

```
df <- read.csv("data.csv", header = TRUE, na.strings=c(""),
                  stringsAsFactors=FALSE)
names(df)
```

```
##  [1] "Year"
##  [2] "Gender"
##  [3] "Educational.Stage"
##  [4] "Field.Of.Interest"
##  [5] "Would.Enroll.In.CS"
##  [6] "Q1"
##  [7] "Q2"
##  [8] "CS.Only.Teaches.To.Use.Software"
##  [9] "CS.Uses.Little.Math"
## [10] "Most.CS.Students.Are.Male"
## [11] "CS.Requires.Knowledge.In.Computers"
## [12] "Higher.Education.Required.To.Work.In.CS"
## [13] "Family.Approves.CS.Major"
## [14] "CS.Has.Low.Employability"
## [15] "CS.Work.Has.Long.Hours"
## [16] "CS.Fosters.Creativity"
## [17] "CS.Is.Prestigious"
## [18] "CS.Provides.Good.Wages"
## [19] "CS.Enables.Interdisciplinary.Experiences"
## [20] "Uses.Computer.At.Home"
## [21] "Uses.Computer.At.Relatives.House"
## [22] "Uses.Computer.At.Friends.House"
## [23] "Uses.Computer.At.School"
## [24] "Uses.Computer.At.Work"
## [25] "Uses.Computer.At.Lan.House"
## [26] "Uses.Computer.At.Library"
## [27] "Uses.Computer.At.Digital.Inclusion.Center"
## [28] "Has.Used.Text.Editor"
## [29] "Has.Used.Image.Editor"
## [30] "Has.Used.Spreadsheet"
## [31] "Has.Used.Database"
```

```
## [32] "Has.Used.Internet"
## [33] "Has.Used.Social.Network"
## [34] "Has.Used.Email"
## [35] "Has.Used.Games"
## [36] "Has.Used.For.Creating.Web.Pages"
## [37] "Has.Used.For.Development"
## [38] "Has.Used.Other.Softwares"
```

Para facilitar o processo de limpeza desses dados, as células do *data frame* que possuem respostas em branco ("") são lidas como NA e assim podem ser removidas.

```
df <- df[complete.cases(df),]
```

Observa-se que alunos que responderam os questinários colocaram seu "Gênero".

```
kable(df %>%
        group_by(Gender) %>%
        dplyr::summarize(total = n()))
```

Gender	total
F	2957
M	9

Como é possível notar, há a presença de 9 meninos que serão removidos da análise.

```
df <- df[df$Gender == 'F',]
```

2. Análise de Características

É importante visualizar a coluna `Would.Enroll.In.CS` pois essa demonstra o interesse do estudante em curso um curso de Ciência da Computação.

Would.Enroll.In.CS	total
Maybe	1134
No	816
Yes	1007

As características importantes para a analise estão presentes na colunas 8 até a 38, as quais são perguntas do questionário que mostram atividades ou hábitos que podem possivelmente aumentar ou diminiur o interesse da estudante nos cursos de computação.

Para essa análise, vamos remover as colunas `Year`, `Gender`, `Educational.Stage`, `Field.Of.Interest`, `Q1`, `Q2` para simplificar a análise.

```
df <- df[, -(1:4), drop=FALSE ]
df <- df[, -(2:3), drop=FALSE ]
```

Preprocessamento

Para simplificar a analise, usa-se a matriz de correlação para identificar quais atributos estão muito relacionados entre si e assim podem ser removidos. Para essa análise, todos os valores devem ser númericos. A função `decide` analisa a resposta em:

1. “Yes” equivale a 2;
2. “Maybe” equivale a 1;
3. “No” equivale a 0

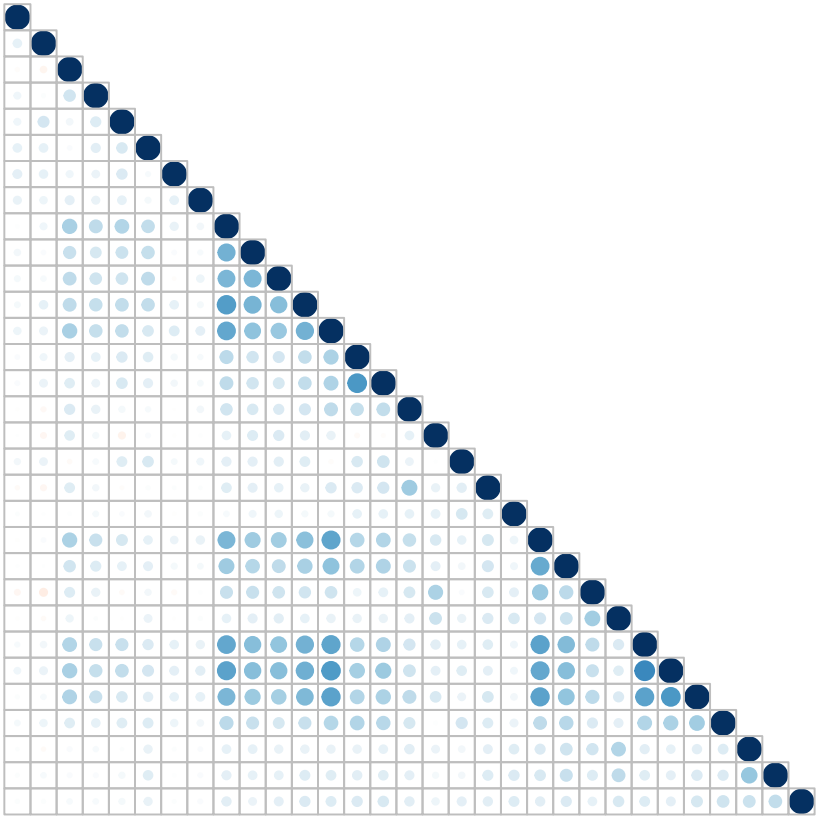
e troca o valor para número. A coluna `Would.Enroll.In.CS` será transformada em factor para análise de categorias.

```
decide <- function(x) {
  switch(x,
    "Yes"= {
      return(2)
    },
```

```
    "Maybe"={
      return(1)
    },
    "No"={
      return(0)
    },
    {
      return(0)
    }
  )
}

for(j in 1:32) {
  for(i in 1:3178) {
    df[i, j] <- decide(df[i, j])
  }
  df[,j] <- as.numeric(df[,j])
}

df$Would.Enroll.In.CS <- factor(df$Would.Enroll.In.CS,
                                levels=c(0,1,2),
                                labels=c("No", "Maybe", "Yes"))
```

Agora a matriz de correlação pode ser calculada:

```
corMatrix <- cor(df[,2:32])

corrplot(corMatrix, method = "circle", cl.pos = "n", tl.pos = "n",
                    type = "lower")
```



Observando o gráfico, existem poucos atributos correlacionados, mas ainda existem.

```
highlyCorrelated <- findCorrelation(corMatrix, cutoff=0.50)
print(names(df)[highlyCorrelated])
```

```
## [1] "Has.Used.Internet"
## [2] "Has.Used.Database"
## [3] "CS.Enables.Interdisciplinary.Experiences"
## [4] "Has.Used.Social.Network"
## [5] "Uses.Computer.At.Digital.Inclusion.Center"
## [6] "CS.Work.Has.Long.Hours"
## [7] "Uses.Computer.At.Relatives.House"
```

Remove-se as colunas com correlação maior que 0.5.

```
df <- df[,-highlyCorrelated]
```

3. Modelo de Treino

Para essa analise, o conjunto de dados será dividido em dois grupos: o grupo de treino e o grupo de testes.

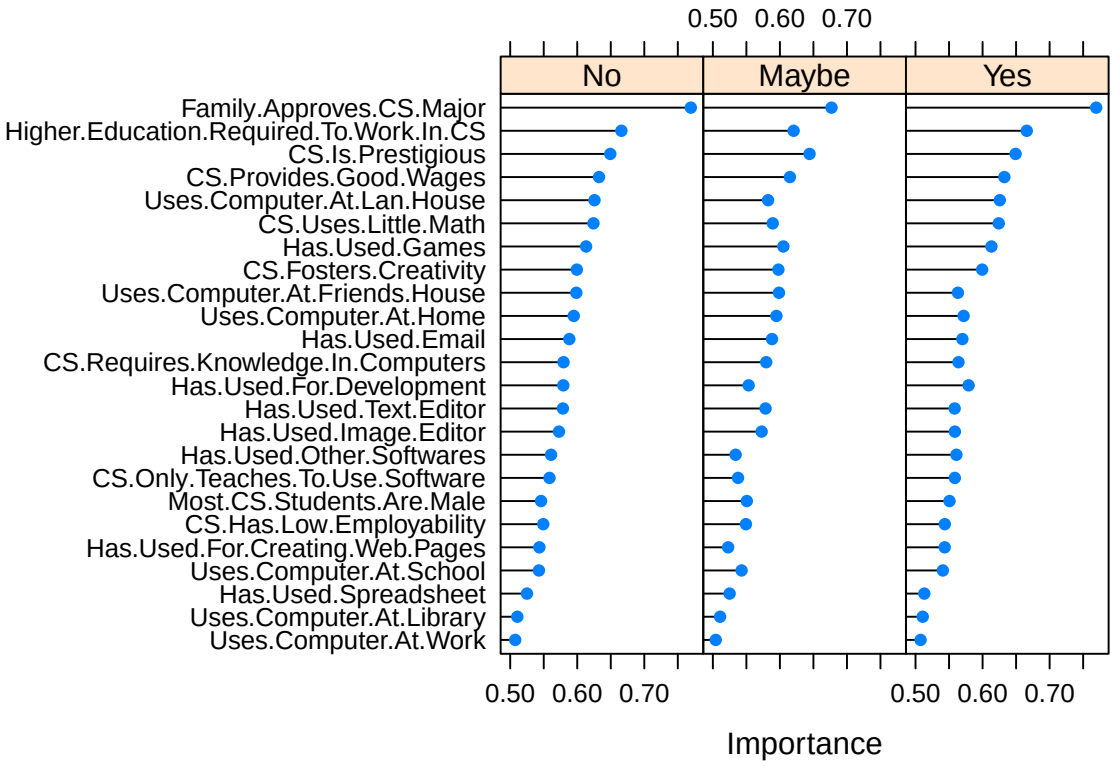
```
trainingIndexes <- createDataPartition(df$Would.Enroll.In.CS,
                                       p=0.85, list=FALSE)
trainingData <- df[trainingIndexes,]
testData <- df[-trainingIndexes,]
```

Usando o conjunto de dados de treino, usa-se a *Support Vector Machine* [1] com uma função polinomial como *kernel* para criar um previsão de como os dados se comportam.

```
trainingParameters <- trainControl(method="repeatedcv", number=10,
                                   repeats=2)

SVModel <- train(Would.Enroll.In.CS ~ .,
                 data = trainingData,
                 method = "svmPoly",
                 trControl= trainingParameters,
                 tuneGrid = data.frame(degree = 1,
                                       scale = 1,
                                       C = 1),
                 preProcess = c("pca","scale","center"),
                 na.action = na.omit
)
```

Com o modelo preparado, usa-se a função `varImp` para descobrir quais colunas tem um impacto maior em cada classificação, ou seja, se o candidato optou por "No", "Maybe" ou "Yes" quando respondeu `Would.Enroll.In.CS`.

```
importance <- varImp(SVModel, scale=FALSE)
plot(importance)
```



Agora, usando o conjunto de dados de teste, uma amostragem é criada tentando prever quantas respostas corretas podem ser alcançadas usando esse modelo.

```
predictions <- predict(SVModel, testData)
cm <- confusionMatrix(predictions, testData$Would.Enroll.In.CS)
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Maybe Yes
##      No    94    31  22
##      Maybe 42    91  47
##      Yes   19    48  82
##
## Overall Statistics
##
##                Accuracy : 0.5609
##                  95% CI : (0.515, 0.6061)
```

```
##     No Information Rate : 0.3571
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3398
##  Mcnemar's Test P-Value : 0.5961
##
## Statistics by Class:
##
##                      Class: No Class: Maybe Class: Yes
## Sensitivity             0.6065       0.5353     0.5430
## Specificity             0.8349       0.7092     0.7938
## Pos Pred Value          0.6395       0.5056     0.5503
## Neg Pred Value          0.8146       0.7331     0.7890
## Prevalence              0.3256       0.3571     0.3172
## Detection Rate          0.1975       0.1912     0.1723
## Detection Prevalence    0.3088       0.3782     0.3130
## Balanced Accuracy       0.7207       0.6222     0.6684
```

Usando o Modelo *Support Vector Machine*, temos uma precisão de 56,51%.

4. Análise exploratória

Preprocessamento

Para manter essa etapa idependente das etapas executadas anteriormente, os dados são relidos dos arquivos `.csv` e limpados novamente.

```
workingdata <- read.csv("data.csv",
                        header = TRUE,
                        stringsAsFactors=TRUE,
                        na.strings=c(""))
numericworkingdata <- read.csv("data.csv",
                        header = TRUE,
                        stringsAsFactors=FALSE,
                        na.strings=c(""))
workingdata <- workingdata[complete.cases(workingdata),]
numericworkingdata <-
  numericworkingdata[complete.cases(numericworkingdata),]
workingdata <- workingdata[workingdata$Gender == 'F',]
numericworkingdata <-
  numericworkingdata[numericworkingdata$Gender == 'F',]
```

Atualmente os dados encontram-se em sua maioria em formato de string, tendo como respostas "Yes", "No" e "Maybe". Para que se torne mais fácil de manipular, iremos transformar tais strings em zeros ("No"), uns ("Yes") e dois ("Maybe").

```
numericworkingdata[,5:38][numericworkingdata[,5:38] == "No"] = 0
numericworkingdata[,5:38][numericworkingdata[,5:38] == "Yes"] = 1
numericworkingdata[,5:38][numericworkingdata[,5:38] == "Maybe"] = 2
```

Os campos de interesse constituem basicamente de "Human Sciences", "Biology-Health Sciences" e "Exact Sciences", nomes relativamente grandes, portanto abreviaremos para 0, 1 e 2 respectivamente.

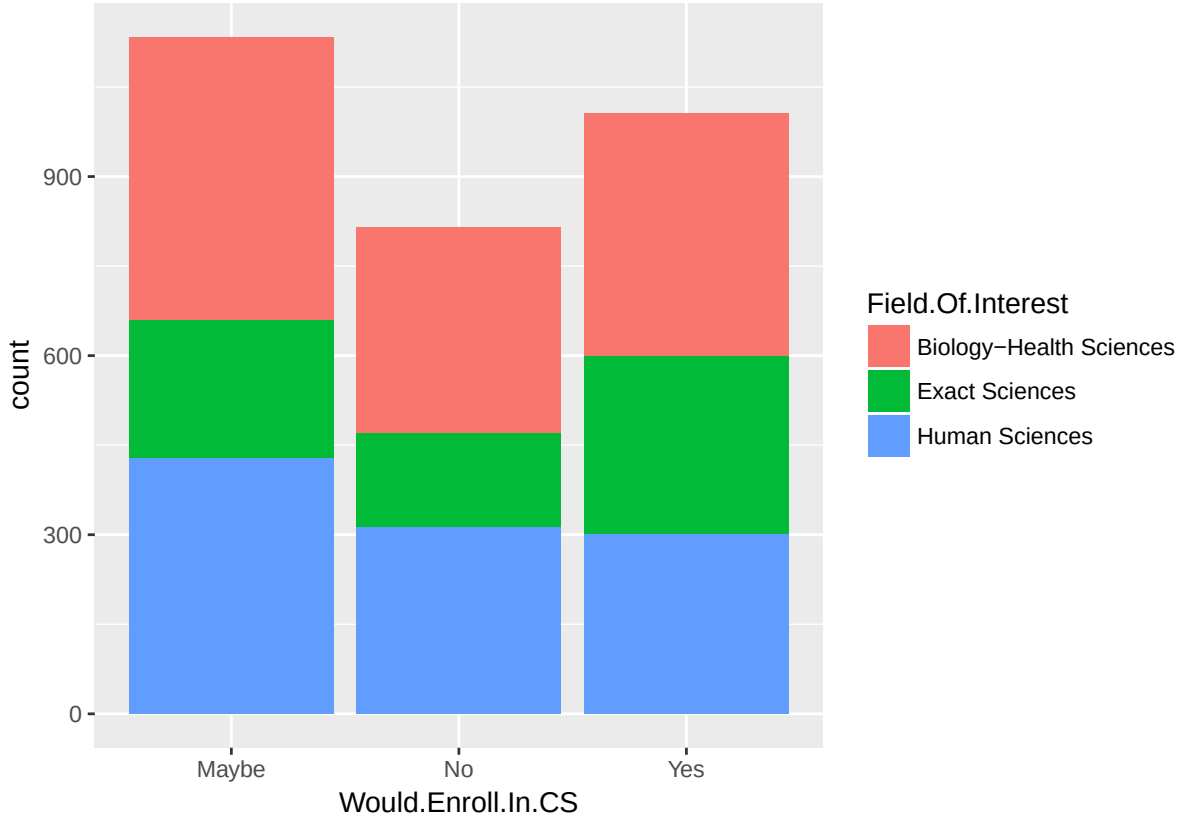
Uma vez transformadas as informações, elas serão convertidas em fatores para melhor manipulá-las.

```
cols <- c(4,5,8:38)
numericworkingdata[cols] <- lapply(numericworkingdata[cols], factor)
```

Agora que o dado encontra-se bem formatado, é possível explorá-lo com maior facilidade.

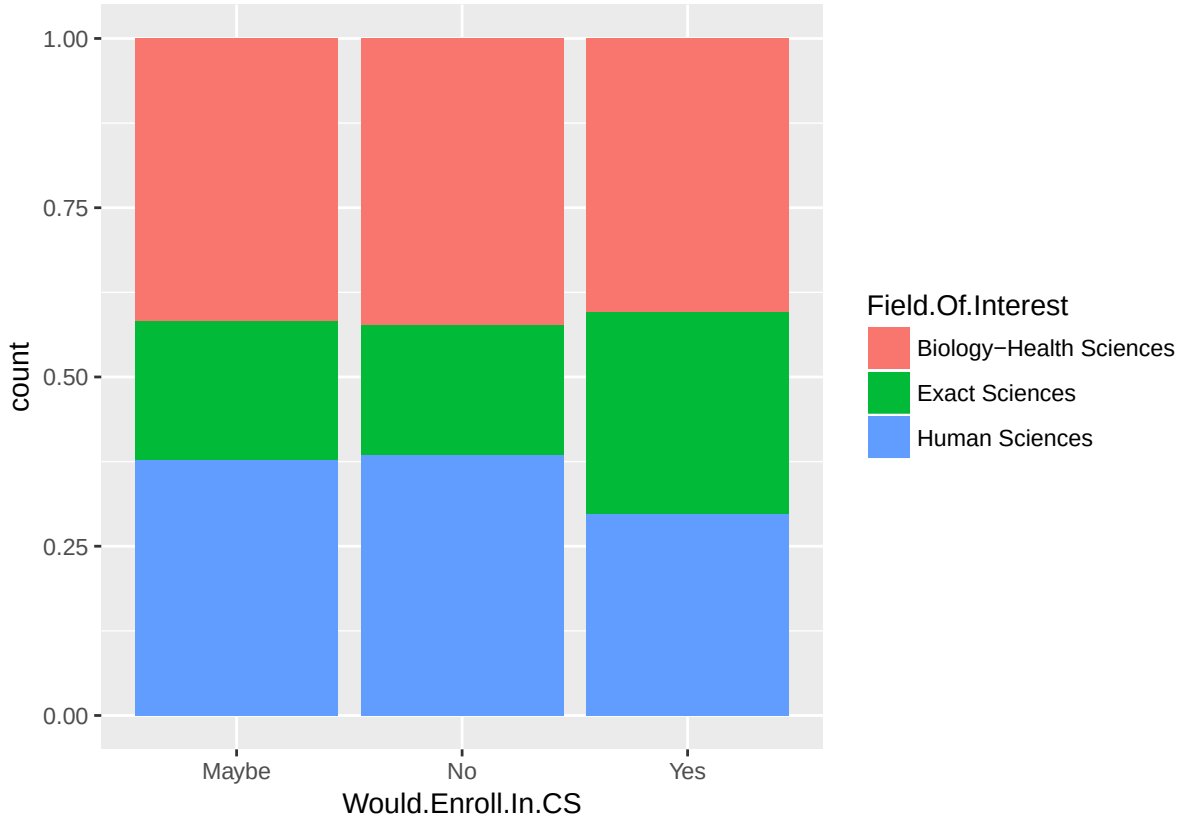
Como é possível notar, enquanto a maior parte das garotas encontra-se entre o não e o talvez, o grupo que entraria na ciência da computação é bem diverso, tendo em sua maioria mulheres interessadas

na área de biologia e não em exatas como se esperaria.



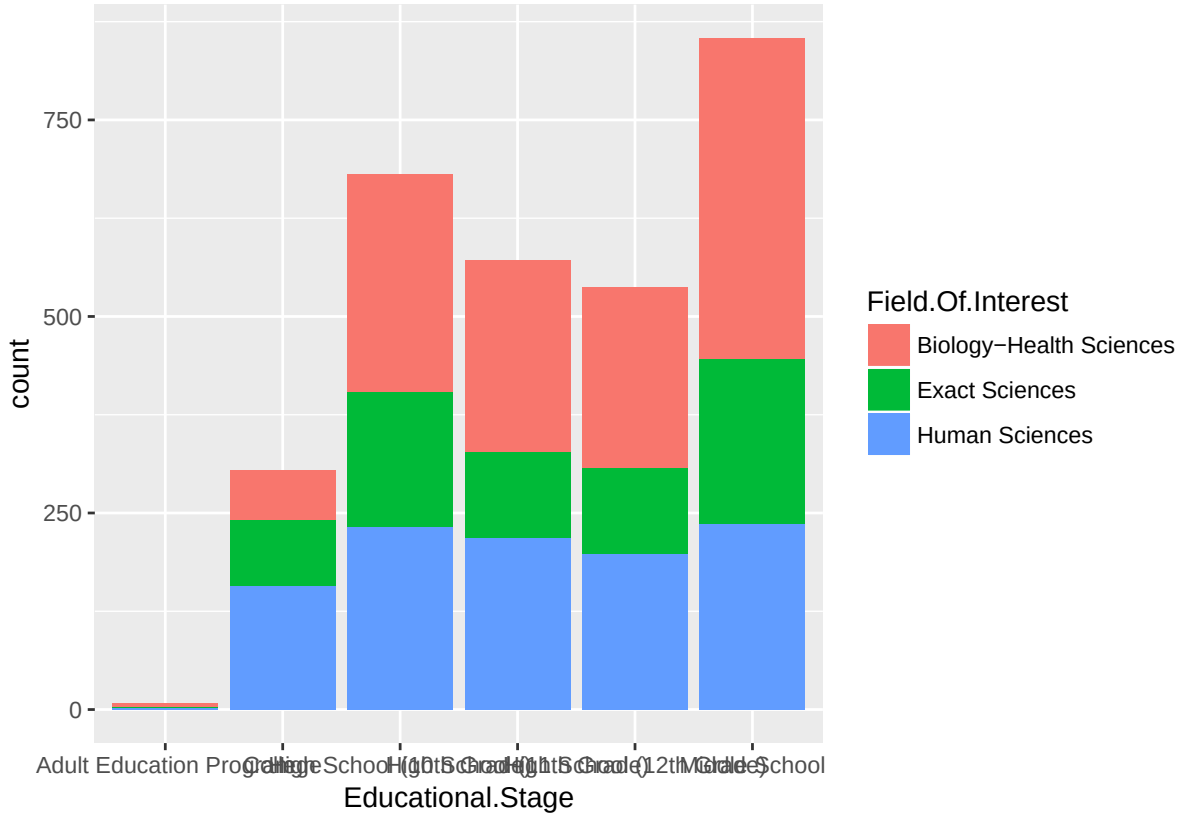

```
##
## Biology-Health Sciences         Exact Sciences         Human Sciences
##                     407                    299                    301
```

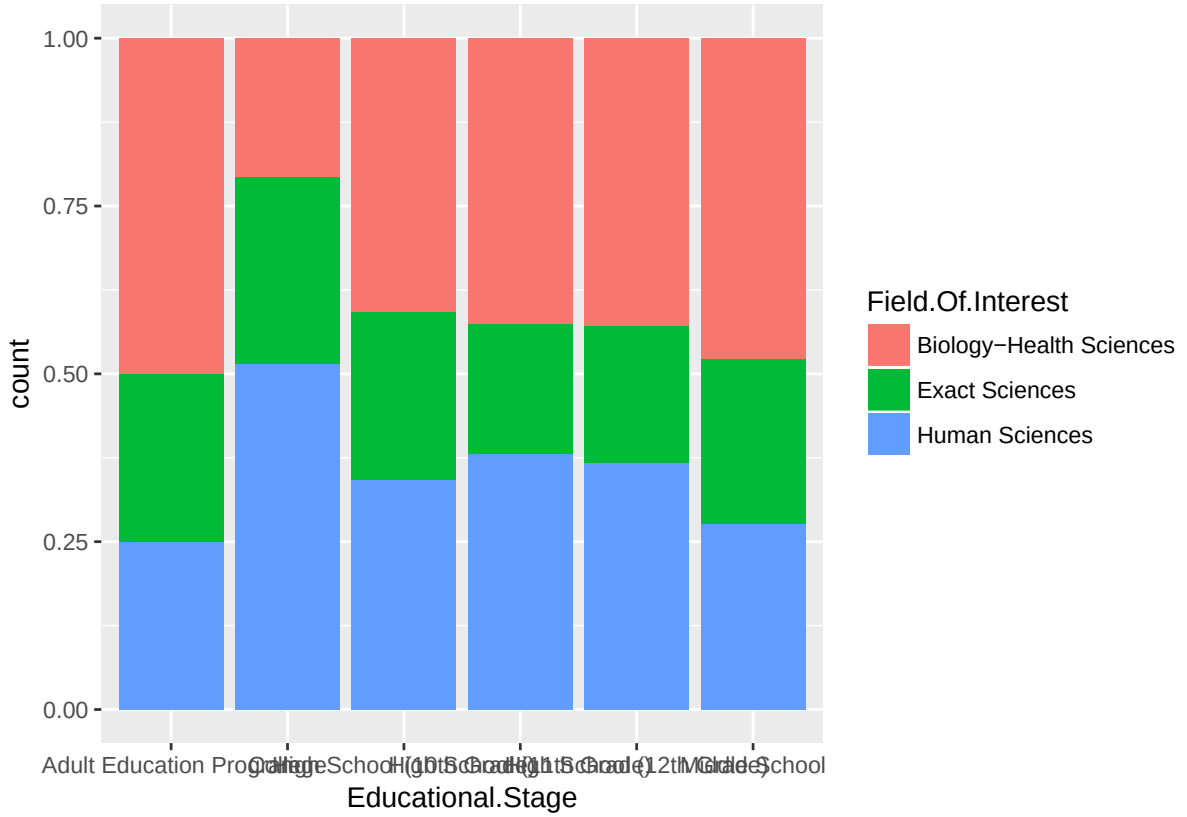
A proporção de interessadas em biologia é consideravelmente maior do que o resto, o que explica o fato de haver tantas mulheres de biologia interessadas em ciência da computação, elas dominam todas as respostas (sim, não e talvez). E de acordo com o gráfico abaixo, que usa proporção em vez de quantidade, tal raciocínio é bem próximo da realidade.



Como é possível notar, Biologia manteve-se próxima em porcentagem em quase todas as respostas enquanto humanas diminuiu e exatas aumentou na resposta afirmativa, o que condiz com a área de interesse.

Em se tratando de quantidade de mulheres que responderam a enquete há um predomínio das que estão no ensino médio. Conforme a escolaridade aumenta, menor é o interesse em exatas, um fenômeno que aponta que a presença feminina está afetada não só na computação.

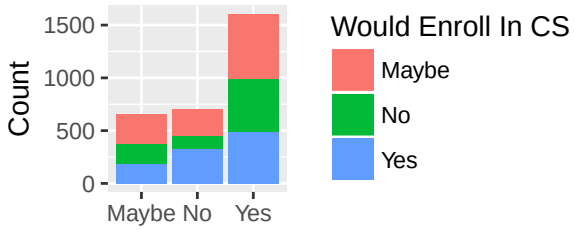

750
500
250
0
count
Adult Education Program
Educational.Stage
Field.Of.Interest
Biology−Health Sciences
Exact Sciences
Human Sciences



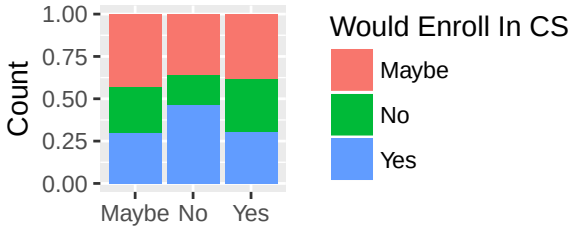
Como há abundância de questões, resolveu-se pegar as que provavelmente mais influenciarão na decisão de seguir carreira na área de CS. As 20 perguntas selecionadas foram as seguintes:

```
perguntas_selecionadas = c("Most.CS.Students.Are.Male",
                           "CS.Requires.Knowledge.In.Computers",
                           "Higher.Education.Required.To.Work.In.CS",
                           "Family.Approves.CS.Major",
                           "CS.Has.Low.Employability",
                           "CS.Work.Has.Long.Hours",
                           "CS.Is.Prestigious",
                           "CS.Provides.Good.Wages",
                           "Uses.Computer.At.Home",
                           "Uses.Computer.At.School",
                           "Uses.Computer.At.Library",
                           "Uses.Computer.At.Digital.Inclusion.Center",
                           "Has.Used.Text.Editor",
                           "Has.Used.Database",
                           "Has.Used.Internet",
```

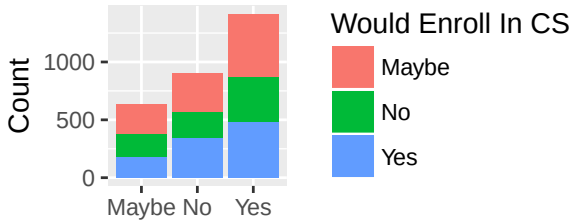
```
                    "Has.Used.For.Creating.Web.Pages",
                    "Has.Used.For.Development")
```



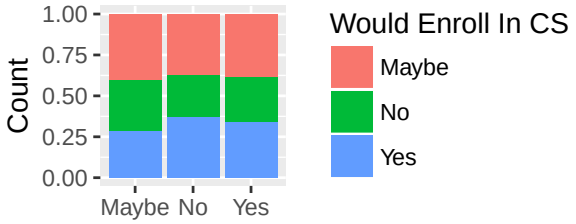
Most CS Students Are Male



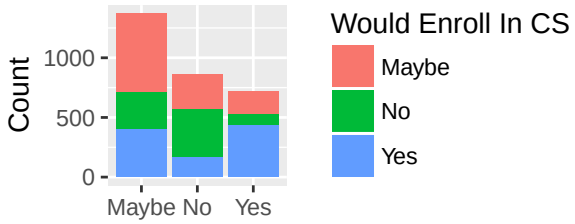
Most CS Students Are Male



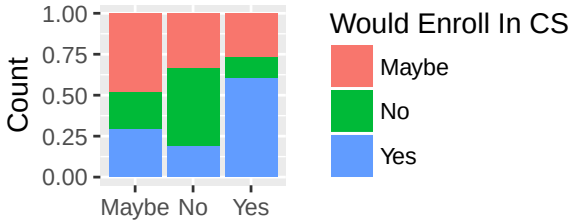
CS Requires Knowledge



CS Requires Knowledge



Family Approves CS Major



Family Approves CS Major

Conclusão

Como a análise realizada nesse relatório, podem-se chegar a duas conclusões:

1. A partitipação da família tem um impacto muito forte para a candidata escolher ou não um curso na área de computação.
2. As perguntas que tem uma importância menor 0.6 podem ter uma abordagem diferente para que em futuras pesquisas possam se aprofundar mais nessa questão e se aproximar mais de uma solução prática.

Referências

[1] https://pt.wikipedia.org/wiki/Máquina_de_vetores_de_suporte acessado em 05/12/2017.